

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**Improved Meta Data Management for Media Content
Objects**

Inventor(s):

**Kirt Debique
David Stewart
Gabriel DeBacker
Nosakhare D. Omoigui**

ATTORNEY'S DOCKET NO. MS1-790US

1 **TECHNICAL FIELD**

2 This invention relates to data structures and data management, and more
3 particularly to meta data management for media content objects.

4
5 **BACKGROUND**

6 Computers are becoming increasingly more powerful while at the same
7 time becoming less costly. This has resulted in the promulgation of computers
8 into many homes and business throughout the world. Along with this increase in
9 computing performance and popularity has also come an increase in the number of
10 areas in which computers are used. Where once computers were used primarily
11 for productivity-based applications (e.g., databases, word processing,
12 spreadsheets, and so forth), a wide range of entertainment applications have
13 become increasingly popular.

14 One such entertainment application is that of media content playback, such
15 as audio (e.g., songs) and audio/video (e.g., movies) playback. For example,
16 computers are often equipped with a CDROM drive that allows the computer to
17 read an audio CD and play the songs on the CD via one or more speakers coupled
18 to the computer. An additional feature that such an entertainment application may
19 provide is the ability to obtain information about a CD (e.g., track names and artist
20 name(s)). This information is commonly referred to as "meta data" corresponding
21 to the CD. The application accesses a database of meta data (e.g., from a remote
22 server) to identify information about the CD (e.g., track names and artist name(s)),
23 and then displays this information to the user.

24 In addition to playing back media content directly from the source CD,
25 some applications also allow songs to be copied from the source CD and stored as

1 separate files on a local hard drive of the computer, such as in an MP3 or WMA
2 format. This process is commonly referred to as "ripping". These locally stored
3 files can then be played back at a subsequent time directly from the local hard
4 drive without the presence of the source CD in the CDROM drive.

5 One current problem with these systems, however, is that meta data is
6 typically associated with only one source (e.g., the source CD). Thus, if the user
7 plays back a song directly from a CD, then the meta data corresponding to that CD
8 is displayed to the user. Similarly, if the user rips that same song to a file on the
9 local hard drive and plays that song back from the file on the hard drive, then the
10 meta data corresponding to that file is displayed to the user. However, any
11 changes made (e.g., by the user) to the meta data corresponding to the CD are
12 displayed to the user only when the song is played back directly from the CD, and
13 any changes made to the meta data corresponding to the file are displayed to the
14 user only when the song is played back from the file. Thus, multiple different sets
15 of meta data are created (one for each source) for each song, requiring any
16 changes to the meta data for a song to be made to each of the multiple sets. This
17 creates an inconsistent and unfriendly user experience.

18 The invention addresses these problems and provides solutions to improve
19 meta data management for media content objects.

20
21 **SUMMARY**

22 Improved meta data management for media content objects is described
23 herein.

24 According to one aspect of the improved meta data management, meta data
25 associated with multiple pieces of content (e.g., songs, movies, other audio/video

1 clips, etc.) stored on multiple pieces of media (e.g., CDs, DVDs, etc.) is
2 maintained in a meta data store. The meta data store also includes meta data
3 associated with other pieces of content stored elsewhere, such as songs stored in
4 files on a local hard drive that have been ripped from a CD or DVD. These other
5 pieces of content are associated with the content on the pieces of media, such as
6 being ripped versions of the same song tracks. This association is maintained in
7 the meta data store, so that whenever a change is made to meta data for one piece
8 of content (e.g., for a track on a CD), then the meta data for the associated piece(s)
9 is also changed (e.g., the ripped version stored in a file on the hard drive).

10 According to another aspect of the improved meta data management, meta
11 data is organized using a disc-centric tree structure, with a disc object being a root
12 object in the tree and multiple children objects (e.g., corresponding to tracks, files,
13 play lists, and so forth) spawning from the disc object. These children can also
14 have their own children object (e.g., file objects). Meta data is maintained for all
15 of these objects, and any changes to the meta data can be readily propagated
16 through the tree to other objects corresponding to the same disc object (the root
17 object).

18 According to another aspect of the improved meta data management, meta
19 data is maintained for both generally available media and user-created media.
20 Meta data for generally available media (e.g., commercially available CDs or
21 DVDs) can be obtained from a remote server. For user-created media, the
22 computer creating the media has access to the meta data associated with the
23 content of the new media (e.g., songs copied from another CD for which meta data
24 is already available). The computer generates a disc identifier for the user-created
25 media and, for each piece of content on the new media, copies the meta data for

1 that content from the meta data associated with the source of the content (e.g., for
2 a song copied from a CD, the meta data associated with that song corresponding to
3 that CD).

4

5 **BRIEF DESCRIPTION OF THE DRAWINGS**

6 The present invention is illustrated by way of example and not limitation in
7 the figures of the accompanying drawings. The same numbers are used
8 throughout the figures to reference like components and/or features.

9 Fig. 1 is a block diagram illustrating an exemplary network environment.

10 Fig. 2 illustrates an exemplary tree structure of media content objects
11 associated with particular media.

12 Fig. 3 illustrates an exemplary database that implements the tree structure
13 of Fig. 2.

14 Fig. 4 illustrates an exemplary return of meta data from a meta data server.

15 Fig. 5 is a flowchart illustrating an exemplary process for rendering media
16 content meta data.

17 Fig. 6 is a flowchart illustrating an exemplary process for updating media
18 content meta data.

19 Fig. 7 is a flowchart illustrating an exemplary process for generating meta
20 data for new media.

21 Fig. 8 illustrates a general exemplary computer environment which can be
22 used to implement the improved meta data management.

1 **DETAILED DESCRIPTION**

2 Fig. 1 is a block diagram illustrating an exemplary network environment
3 100. Environment 100 includes a client computer 102, remote server computers
4 104 and 106, and a network 108. Network 108 represents any of a wide variety of
5 conventional data communications networks. Network 108 may include public
6 portions (e.g., the Internet) as well as private portions (e.g., an internal corporate
7 Local Area Network (LAN) or a home network), as well as combinations of public
8 and private portions. Network 108 may be implemented using any one or more of
9 a wide variety of conventional communications media including both wired and
10 wireless media. Any of a wide variety of communications protocols can be used
11 to communicate data via network 108, including both public and proprietary
12 protocols. Examples of such protocols include TCP/IP, IPX/SPX, NetBEUI, etc.

13 Computers 102, 104, and 106 represent any of a wide range of computing
14 devices, and each device may be the same or different. By way of example,
15 devices 102 – 106 may be desktop computers, multiple-processor fileservers or
16 workstations, laptop computers, handheld or pocket computers, personal digital
17 assistants (PDAs), cellular phones, Internet appliances, consumer electronics
18 devices, gaming consoles, and so forth.

19 Client computer 102 includes a local hard drive 110, a system memory
20 (Random Access Memory (RAM)) 112, a disc drive 114 for removable media 116,
21 and a central processing unit (CPU) 118. CPU 118 executes an operating system
22 120, which manages operation of client computer 102 and provides support for
23 input/output operations (e.g., user inputs, audio and/or video input and playback,
24 data communications with other computers via network 108, and so forth). CPU
25 118 also executes a content player application 122 from memory 112. Content

1 player application 122 provides a user interface allowing a user of computer 102
2 to select various pieces of media content (also referred to as simply "content") for
3 playback. Content player application 122 also communicates audio data to
4 speakers of computer 102 (optionally via operating system 120) for playing to the
5 user, as well as communicates video data to a display device of computer 102 for
6 playback of video data. Additionally, content player application 122 may also
7 present (e.g., display) to the user meta data corresponding to content being played.
8 This meta data is information describing different aspects or characteristics of the
9 content being played, and/or information related to the content being played. A
10 meta data manager component 124 of content player application 122 manages the
11 storage and modification of meta data, as discussed in more detail below.
12 Although illustrated as separate components, content player 122 and operating
13 system 120 may alternatively be incorporated into the same component (e.g.,
14 content player 122 may be one of multiple modules of operating system 120).

15 Content player application 122 can play pieces of content directly from
16 media 116 inserted into disc drive 114. A piece of content as used herein refers to
17 a segment of data that can be played (e.g., to a user on a computer). The exact
18 nature of a piece of content on media 116 is dependent on the type of media as
19 well as the type of content. For example, for an audio CD, each track of audio
20 data (typically a single song) would be a piece of data. For a movie DVD, the
21 entire movie may be one piece of content, a set of outtakes another piece of
22 content, and a movie preview a third piece of content.

23 Media 116 can be any of a wide variety of conventional media. As used
24 herein, media 116 is discussed primarily with reference to being a compact disc
25 (CD) or digital versatile disc (DVD), including CDs and DVDs that are read only,

1 rewriteable, or recordable. Alternatively other types of media can be used, such as
2 other types of optical discs, magnetic disks, magnetic tapes, solid state storage
3 devices, and so forth. It is to be appreciated that the nature of drive 114 also
4 varies based on the type of media 116 being used (e.g., an optical disc drive, a
5 magnetic tape drive, and so forth).

6 Additionally, client 102 may access other media in a manner analogous to
7 accessing media 116. For example, media 116 may be inserted into a disc drive of
8 another computer (or CD "jukebox") and accessed by client 102 via network 108.
9 By way of another example, an Internet radio station may make tracks of data,
10 analogous to tracks of data on a CD, accessible to client 102.

11 In addition to being used as a direct source for content playback, media 116
12 may be used as the source for content that is stored as one or more files on hard
13 drive 110. Content player application 122 can manage the copying (ripping) of
14 content from media 116 to a file 126 of hard drive 110, or alternatively another
15 component such as operating system 120 or a component on another computer
16 (not shown) may manage the copying. Each of these ripped files is another
17 version of the corresponding piece of content from media 116. Each of the
18 content files 126 typically stores a single piece of content, and may use any of a
19 wide variety of public and/or proprietary formats, such as MPEG Audio Layer 3
20 (MP3), Windows Media audio file (WMA), and so forth.

21 During operation, meta data manager 124 maintains a record (explicitly or
22 implicitly) of different media pieces 116, such as different CDs, that are inserted
23 into disk drive 114. When coupled to network 108, meta data manager 124
24 communicates with server 104 and/or 106 to retrieve meta data corresponding to
25 the different media 116 that has been inserted into disc drive 114. Meta data

1 manager 124 stores this received meta data locally (e.g., as local meta data store
2 128 on hard drive 110) for subsequent use by content player application 122.

3 Content player application 122, including meta data manager 124, provides
4 an enhanced content playback experience by presenting, to the user, meta data
5 corresponding to various pieces of content. The meta data can be displayed while
6 the content is being played, as well as at other times (e.g., allowing the user to
7 look back through content that he or she has previously played). Meta data
8 manager 124 obtains (from server 104 and/or 106) meta data for each piece of
9 content (e.g., each song) on a particular piece of media (e.g., each CD) that has
10 been inserted into disc drive 114. This meta data is obtained even though the user
11 may not have actually played the corresponding content – the mere insertion of the
12 media is sufficient. In the illustrated example meta data manager 124 obtains only
13 the meta data for pieces of media it is aware of (e.g., those that have been inserted
14 into disc drive 114). Alternatively, other meta data may also be obtained and
15 stored in local meta data store 128. For example, a user may identify a preferred
16 artist and all meta data available for all of that artist's CDs may be obtained and
17 stored in local meta data store 128.

18 Meta data manager 124 associates meta data with the particular pieces of
19 removable media 116 and further associates all pieces of content (e.g., songs,
20 movies, and so forth) that are on or originate from particular media 116 with that
21 particular media 116. Thus, meta data for a particular song from a particular disc
22 is associated with that song, regardless of whether it is played to the user directly
23 from media 116 or from one of content files 126 that was ripped from media 116.

24 In the illustrated example, the meta data is stored at servers 104 and 106
25 and is available from a "disc-centric" point of view. This "disc-centric" point of

1 view organizes meta data on a per-disc basis, with tracks on that disc and files
2 ripped from those tracks corresponding to the particular disc. A disc identifier is
3 used for tracking the meta data for the disc. In the case of CDs, CDs typically
4 store songs as different tracks on the CD, and each track is made up of multiple
5 "frames" of audio data. In one implementation, each frame of audio data is 1/75 of
6 a second. Typical audio CDs store table of contents (TOC) information, which
7 refers to: the total number of tracks on the disc, the starting frame offset of each
8 track, and the total number of frames on the disc. The disc identifier is generated
9 by summing together this table of contents information (the total number of tracks
10 on the disc, the starting frame offset of each track, and the total number of frames
11 on the disc). It should be noted that generating a disc identifier in this fashion
12 does not guarantee that each disc identifier is unique. However, it does provide an
13 almost-unique value (that is, it is possible for two different CDs to have the same
14 disc identifier, but highly unlikely).

15 Alternatively, the disc identifier may be generated in different manners for
16 CDs, as well as different manners for different types of media. For example, a
17 disc identifier for a DVD may be generated based on a CRC 64 (64-bit Cyclical
18 Redundancy Checking) of portions of the DVD. In one implementation, these
19 portions are the first parts (e.g., 64Kb) of the `video_ts.ifo` and the `vts_01_0.ifo` on
20 a DVD disc.

21 When meta data manager 124 desires meta data for a particular disc,
22 manager 124 communicates the disc identifier to one of servers 104 and 106. The
23 receiving server 104 or 106 uses the disc identifier as an index into the meta data
24 stored at the server. If the disc identifier matches a disc identifier at the server,
25

1 then the meta data associated with that matching server is returned to meta data
2 manager 124.

3 Servers 104 and 106 can store public and/or private meta data. Public meta
4 data refers to data that is generally available to the public as a whole (although
5 possibly subject to licensing fees or other restrictions). Private meta data refers to
6 data that is intended for only a particular one or more users. For example, a user
7 may generate his or her own meta data and communicate it to a server as private
8 meta data. Subsequent accesses to the server by the user (the identity of which can
9 be verified in any of a variety of conventional manners, such as a user name and
10 password) allow this meta data to be received. By storing such private meta data
11 on a server, a user can generate his or her own data and have that data "follow"
12 him or her to different locations. For example, a user may generate private meta
13 data from his or her home computer and store it on a remote server. The user can
14 then access that server from another location (e.g., his or her portable audio
15 device, an Auto PC, and so forth) and have that previously generated private data
16 available at the other location. Thus, if the user generates special meta data for a
17 CD on his or her home computer, when the user inserts the CD into his or her
18 Auto PC, the same special meta data can be retrieved from the server and rendered
19 by the Auto PC.

20 Content player 122 maintains a tree-structure of objects associated with
21 particular media 116. In the illustrated examples, this tree-structure is a "disc-
22 centric" structure placing a disc identifier as the root node and then one or more
23 levels of children depending from the root node.

24 Fig. 2 illustrates an exemplary tree structure 148 of media content objects
25 associated with particular media. Each object in tree structure 148 can have meta

1 data associated with it. Two root node objects 150 and 152 are illustrated, each
2 corresponding to a particular optical disc. Root object 150 has three children
3 objects 154, 156, and 158, each of which is one track on the disc corresponding to
4 object 150. Additionally, track object 154 has two children 160 and 162, each of
5 which is a file that has been ripped to a local hard drive, and each of which is a
6 ripped version of the track corresponding to track object 154. Furthermore, track
7 object 158 has one child 164, which is a file that is stored on the local hard drive
8 and is a ripped version of the track corresponding to track object 158. The track
9 corresponding to track object 156 has not been ripped to any files on the local hard
10 drive, and thus track object 156 does not have any children.

11 Similarly, two track objects 166 and 168 are child objects from root disc
12 object 152. Track object 166 has a file child object 170, while track object 168
13 has no children.

14 The tree structure of Fig. 2 allows meta data manager 124 to readily
15 propagate meta data, as well as changes to meta data, through the various objects
16 associated with a particular disc (or other media). For example, if a user desired to
17 correct a typographical error that occurred in the name of a song corresponding to
18 track 154, given the relationships identified in Fig. 2 the association of ripped file
19 objects 160 and 162 to track object 154 can be readily identified, and the
20 correction also made in the name of the song corresponding to each of the ripped
21 file objects 160 and 162.

22 The discussions herein refer primarily to a disc-centric point of view as
23 illustrated in Fig. 2. Alternatively, the database may be structured from other view
24 points. For example, rather than the disc object as a root node there may be an
25 "anthology" object as the root node, with each of the disc objects 150 and 152

1 being children of the anthology object. An anthology may include, for example, a
2 set of multiple CDs corresponding to the same artist, genre, time period, etc. In
3 this situation, any meta data changes can be propagated across multiple CDs (e.g.,
4 a performance date for the anthology may be changed and this change propagated
5 to each CD in that anthology). By way of additional examples, other points of
6 view may have artist names as root nodes, genres as root nodes, and so forth.

7 Additionally, it should be noted that even though a disc-centric point of
8 view is used, an object can be a child of objects that are (or lead to) two different
9 root node objects. For example, as discussed in more detail below, a new CD may
10 be created by a user that includes tracks from another source CD. In this situation,
11 the track objects can be linked to parent root node objects that are the root nodes
12 for both the new CD as well as the source CD.

13 The tree structure of Fig. 2 can be implemented using any of a wide variety
14 of data structures.

15 Fig. 3 illustrates an exemplary database 200 that implements the tree
16 structure of Fig. 2. Database 200 is illustrated including an object table 202, a
17 relationships table 204, and a properties table 206. Object table 202 identifies
18 multiple objects in database 200, relationships table 204 maintains a mapping of
19 objects to each other in a specific order, and properties table 206 maintains a set of
20 properties (meta data) for objects in database 200. Fig. 3 illustrates database 200
21 populated with example data for purposes of explanation.

22 Object table 202 identifies the objects that are in database 200. As used
23 herein, an object refers to an entity that can have "children" (that is, one or more
24 other related objects) and properties associated with it. For example, an object
25 may be a disc, a track on a disc, a file ripped from a track on a disc, a play list

1 identifying multiple tracks and/or files, an artist (corresponding to a disc, a track, a
2 file, etc.), a genre (corresponding to a disc, a track, a file, etc.), and so forth. In the
3 illustrated example of Fig. 3, several different objects are identified.

4 Object table 202 includes two fields: a type field and an ID field. For each
5 entry in object table 202, the type field identifies what type of object is described
6 in the entry and the ID field identifies a unique identifier (unique within database
7 200) of the object. In one implementation, the type field includes a globally
8 unique ID that corresponds to the type of object described in the entry. The ID for
9 an object is used to identify the object in the relationships table 204 and properties
10 table 206. The ID for each object can be assigned in any of a wide variety of
11 manners, and in the illustrated example each object is assigned an ID that is the
12 order the object is entered into object table 202 relative to the other objects in the
13 table. In the illustrated example of Fig. 3, a CD is assigned the ID of "1", three
14 CD tracks are assigned the ID's of "2", "3", and "4", respectively, a file is assigned
15 the ID of "5", and so forth.

16 Additional fields may also optionally be included in object table 202. For
17 example, a "description" field may be added allowing information describing the
18 entry to be included (e.g., a brief description of what object the globally unique ID
19 refers to, such as "compact disc").

20 Relationships table 204 identifies relationships between different objects in
21 database 200. Each entry in relationships table 204 identifies a parent/child
22 relationship between two objects identified in object table 202. Relationships
23 table 204 includes four fields: a parent field, a child field, a type field, and an
24 order field. The parent field identifies one of the objects in object table 202 by its
25 ID, while the child field identifies another of the objects in object table 202 by its

1 ID. The parent and child fields identify a parent/child relationship between these
2 two identified IDs. For example, the first entry in relationships table 204 indicates
3 that the object with an ID of 2 is a child of the object with an ID of 1. Thus, in
4 conjunction with object table 202, this first entry indicates that the track with an
5 ID of 2 is a child of the CD with an ID of 1 (in other words, the track with an ID
6 of 2 is a track on the CD with an ID of 1).

7 The type field in relationships table 204 indicates what type of object the
8 object referenced in the child field of the entry is. For example, the first entry
9 indicates that the object with an ID of 2 is a track object. This is a duplicate of the
10 information stored in object table 202, and is included to increase efficiency in
11 accessing database 200. For example, to identify all of the tracks with the same
12 parent, a search can be performed based on the parent and type fields of
13 relationships table 204. Alternatively, the type field may not be included in
14 relationships table 204.

15 The order field in relationships table 204 identifies the order in which each
16 entry for each parent object is added to relationships table 204. The first
17 relationship entered in table 204 for a particular parent object has an order of "1",
18 the second has an order of "2", and so forth.

19 Additional fields may also optionally be included in relationships table 204.
20 For example, an additional type field may be included that indicates what type of
21 object the object referenced in the parent field of the entry is. By way of another
22 example, a User ID field may be included that identifies which user inserted the
23 CD that resulted in the relationship for that entry. The User ID field may be used,
24 for example, to allow meta data for multiple different users to be stored in the
25

1 same database 200, but also allow a particular user to access only the meta data
2 that corresponds to discs he or she inserts into the disc drive.

3 Properties table 206 stores meta data for the objects identified in object
4 table 202. Properties table 206 includes four fields: an ID field, a property (prop)
5 ID field, a "modified by" field, and a value field. For each entry, the ID field for
6 the entry identifies one of the objects in object table 202 by its ID. The property ID field for
7 the entry identifies a particular property that the entry corresponds to. Multiple
8 pieces of meta data can be associated with an object (e.g., the title of a CD, the
9 artist of a CD, a performance date, and so forth). Multiple property ID values are
10 established that uniquely identify the different possible pieces of meta data, and
11 for each entry the property ID field includes one of those ID values.

12 The "modified by" field identifies the source of the meta data for the entry.
13 The modified by field may indicate a particular service that is the source of the
14 meta data (e.g., one of servers 104 or 106 of Fig. 1), or alternatively that the user is
15 the source of the meta data. The value field identifies the meta data for the entry.
16 Any of a variety of meta data can be stored in the value field, and optionally may
17 be modified by the user. For example, properties table 206 indicates that the
18 object having an ID of 3 (which is a CD track) has a user-set value of "1812
19 Overture".

20 Although illustrated as a single field, the value field may optionally be
21 multiple data-dependent fields (e.g., a numeric value field, a text value field, a date
22 value field, and a binary value field).

23 The meta data illustrated in database 200 can be received from a remote
24 server, such as server 104 or 106 of Fig. 1. This meta data can be returned to the
25 client computer implementing database 200 in any of a wide variety of formats.

1 Fig. 4 illustrates an exemplary return of meta data from a meta data server. The
2 meta data 220 illustrated in Fig. 4 is returned in an extensible markup language
3 (xml) format. Alternatively, other public and/or proprietary formats can be used.
4 As illustrated, meta data 220 indicates that the name of the disc is "Billboard Top
5 Hits: 1983", the author is "Various Artists", the release date for the disc is "1992",
6 the genre of the disc is "rock", the style of the disc is "Pop/Rock", and the rating of
7 the disc is "8". Additionally, there is no copyright information in meta data 200
8 for the disc, indicating that the remote server does not have the copyright meta
9 data for the disc. Additionally, the tracks of the disc are identified in meta data
10 220 in order, so the first track on the disc has the name "Africa" and the author
11 "Toto", the second track on the disc has the name "Stray Cat Strut" and the author
12 "Stray Cats", and so forth.

13 Returning to Fig. 3, the various tables in database 200 allow changes to the
14 meta data in the database to be automatically propagated to other related objects.
15 For example, assume that the meta data received from a server 104 or 106 of Fig.
16 1 for the object having an ID of 3 (a CD track) indicates that the title is
17 "Tchaikovsky: 1812 Overture". Further assume that the user changes the title for
18 that track to be simply "1812 Overture". The meta data manager then checks
19 relationships table 204, which indicates that the object having an ID of 3 has a
20 child with an ID of 5 (a file, as indicated in object table 202). The meta data
21 manager thus also changes the title data, in properties table 206, for the object
22 having an ID of 5 to be the new title. Thus, regardless of whether the user
23 subsequently selects playback of Tchaikovsky's 1812 Overture from the CD or the
24 file, the meta data stored in database 200 indicates that the title is simply "1812
25 Overture".

1 Additional fields may also optionally be included in properties table 206.
2 For example, a type field may be included (analogous to the type field in
3 relationships table 204) that identifies what type of object the object referenced in
4 the ID field is. By way of another example, properties for particular objects may
5 be organized into different property sets, such as pre-defined properties and user-
6 defined properties. In this situation, an additional field may be included in
7 properties table 206 that identifies which property set the entry refers to.

8 In addition to tables 202, 204, and 206, one or more additional tables may
9 optionally be included in database 200. In one implementation, these additional
10 tables include a property set types table, a property types table, a users table, and
11 an object values table. Each entry in the property set types table identifies a
12 globally unique ID for a property set, a corresponding ID for that property set, and
13 a description of that property set. Each entry in the property types table identifies
14 one of the property set ID's from the property set types table, a number associated
15 with the property ID, a data type associated with the property (e.g., a numeric
16 value, a text value, a date value, or a binary value), and a description of the
17 property. Each entry in the users table identifies a user ID and corresponding user
18 name of the computer on which database 200 is implemented. Each entry in the
19 object values table identifies an object ID from object table 202, an instance
20 number of the object in the object values table, and optionally an identification of
21 the user that created the object (e.g., a sequential ID based on the user's logon
22 name).

23 Exemplary objects and properties are illustrated in Fig. 3, although a wide
24 range of objects and associated properties may be included in database 200. The
25 following tables illustrate an exemplary set of objects and the properties that may

1 be associated with those objects in accordance with certain implementations.
2 Additionally, the attached Appendix identifies a set of interfaces and functions that
3 can be used in accordance with certain implementations to create, manage, and
4 modify different schemes (e.g., different databases 200 of Fig. 3), and objects
5 within those schemes.

6

7 Object Types

Object Name	Description
Genre	The genre or category of the content.
CDGenre	The genre or category of a CD.
CDArtist	The artist of a CD.
CDAlbum	A CD album.
Artist	The artist of content.
UserCollection	A user-defined collection of content.
Album	An album.
AlbumTrack	A track of an album.
AlbumPlaylist	A play list for an album.
Playlist	A play list of content.
PlaylistFromAlbum	A play list of ripped tracks associated with an album.
Track	A track of content on media.
Batch	A list of disc(s) for which meta data has not been downloaded yet (allowing for batch processing of downloads when network connection is made).
Link	A URL reference and friendly name about an object. For example, "More about Sting" would be a friendly name and the URL reference would be "http://....sting.com".
Picture	An image corresponding to content.
ChildListEntry	A list of the children tracks in a play list.
DVD	A DVD.
DVDTitle	The title of a DVD.
DVDChapter	A chapter of a DVD.
Factoid	One or more facts or comments associated with an album or content. Different factoids may be tied to different frames or other segments of the album or content.

1 **Genre Object**

Property Name	Description
Name	The name of the genre.
NumMediaChildren	The total number of children of the object that are media content.
NumAudioMediaChildren	The total number of children of the object that are audio media content.
NumVideoMediaChildren	The total number of children of the object that are video media content.

7 **CDGenre Object**

Property Name	Description
Name	The name of the genre or category of a CD.

11 **CDArtist Object**

Property Name	Description
Name	The name of an artist of a CD.
Role	The role of an artist of a CD (e.g., lead singer, backup singer, keyboard, etc.).

15 **CDArtist Object**

Property Name	Description
Title	The title of the CD.
Copyright	Copyright information for the CD.
Label	The recording company label that produces the CD.
ReleaseDate	The date of release of the CD.
Rating	A rating for the CD.
RatingOrg	An organization that rated the CD.
CDID	A hashed value of the disc identifier, allowing for faster lookups.
TOC	The disc identifier (based on the table of contents information) for the CD.
LastPlayed	The last time a track of the CD was played on the computer.
Artist	The artist of the CD.

1	Genre	The genre or category of the CD.
2	NumMediaChildren	The total number of children of the object that are media content.
3	NumAudioMediaChildren	The total number of children of the object that are audio media content.
4	NumVideoMediaChildren	The total number of children of the object that are video media content.

5

6 Artist Object

7 Property Name	8 Description
9 Name	10 The name of an artist of a CD.
11 Role	12 The role of an artist of a CD (e.g., lead singer, backup singer, keyboard, etc.).
13 NumMediaChildren	14 The total number of children of the object that are media content.
15 NumAudioMediaChildren	16 The total number of children of the object that are audio media content.
17 NumVideoMediaChildren	18 The total number of children of the object that are video media content.

19 UserCollection Object

20 Property Name	21 Description
22 Name	23 The name of the user-defined collection of content.

24 Album Object

25 Property Name	26 Description
27 Title	28 The title of the album.
29 Copyright	30 Copyright information for the album.
31 Label	32 The recording company label that produces the album.
33 ReleaseDate	34 The date of release of the album.
35 Rating	36 A rating for the album.
37 RatingOrg	38 An organization that rated the album.
39 CDID	40 A hashed value of the disc identifier for the album, allowing for faster lookups.
41 TOC	42 The disc identifier (based on the table of contents information) for the album.

1	LastPlayed	The last time content from the album was played on the computer.
2	Artist	The artist of the album.
3	Genre	The genre or category of the album.
4	NumMediaChildren	The total number of children of the object that are media content.
5	NumAudioMediaChildren	The total number of children of the object that are audio media content.
6	NumVideoMediaChildren	The total number of children of the object that are video media content.

AlbumTrack Object

Property Name	Description
Title	The title of a track of the album.
Lyrics	The lyrics for the content of the track.
LengthInMilliseconds	The length of the track (in milliseconds).
LastPlayed	The last time the track was played on the computer.
Album	The name of the album the track is on.
Artist	The artist of the track.
Genre	The genre or category of the track.
OriginalIndex	The original position of the track on the album it came from.
RipFilename	The name of a file ripped from the track.
Enabled	Whether the track is marked as "playable" on the album. Non-enabled tracks appear "grayed out" in the user interface and are skipped over rather than played when the album is played.
TOC	The disc identifier (based on the table of contents information) of the original CD the track was on.
PlayCount	The number of times the track has been played on the computer.
LastEditedBy	An identification of the last entity (either the name of the meta data provider or "user") to edit any property associated with the object.
Provider	The name of the original meta data provider.
Composer	The composer of the album.
CoverArtSmall	A small graphic of the album's cover art.
CoverArtLarge	A larger graphic of the album's cover art.

1 AlbumPlaylist Object

2 Property Name	3 Description
4 Name	5 The name of the album play list.
6 Copyright	7 Copyright information for the album.
8 Label	9 The recording company label that produces the album.
10 ReleaseDate	11 The date of release of the album.
12 Rating	13 A rating for the album.
14 Style	15 The style of the content in the play list.
16 CDID	17 A hashed value of the disc identifier, allowing for faster lookups.
18 TOC	19 The disc identifier (based on the table of contents information) for the album.
20 LastPlayed	21 The last time a track in the play list was played on the computer.
22 Artist	23 The artist of the album.
24 Genre	25 The genre or category of the album.
26 Parent	27 The parent of the object.
28 LastPlaylistEntryID	29 The ID of the last entry in the play list.
30 LastEditedBy	31 The user that last edited the play list.
32 Composer	33 The composer of the album.
34 TotalDuration	35 The total play time of the content on the play list.
36 BuyNow	37 Information on how to purchase the album.
38 MoreInfo	39 A URL to a web page with more information about the object.

17 PlaylistProperties Object

18 Property Name	19 Description
20 Name	21 The name of the play list.
22 NumMediaChildren	23 The total number of children of the object that are media content.
24 NumAudioMediaChildren	25 The total number of children of the object that are audio media content.
26 NumVideoMediaChildren	27 The total number of children of the object that are video media content.
28 LastPlaylistEntryID	29 The ID of the last entry in the play list.
30 Attributes	31 A set of one or more flags that specify different play list attributes, such as a "deleted" indicating the play list is marked for deletion.
32 CreationTime	33 The time the play list was created.

1	LastModifiedTime	The last time the play list was modified.
2	TotalDuration	The total playtime of content on the play list.
3	Author	The author of the play list.
4	Genre	The genre or category of content on the play list.
5	Copyright	Copyright information for the content on the play list.

PlaylistFromAlbum Object

Property Name	Description	
6	Name	The name of the play list.

Track Object

Property Name	Description	
7	Filename	The filename of the track on the hard drive.
8	Title	The title of the track.
9	Subtitle	Subtitle information for the track.
10	Copyright	Copyright information for the track.
11	FileType	Type of file (e.g., MP3 or WMA).
12	Time	The time the file was created.
13	Date	The date the file was created.
14	Language	The language the content of the track is in.
15	MediaType	The type of media the track is on.
16	PartInSet	Indicates which part of a set the track is on (e.g., which disc of a multiple-disc set).
17	EncodedBy	Name of the person who ripped the track.
18	Publisher	The publisher of the media the track is on.
19	TrackNumber	The number of the track on the media.
20	RecordingDates	The recording date(s) of the track.
21	NetRadioStation	The name of an Internet-accessible radio station where the track is received from.
22	NetRadioOwner	The owner of an Internet-accessible radio station where the track is received from.
23	Year	Year of original recording of the track.
24	BeatsPerMinute	The number of beats per minute in the content of the track.
25	MusicalKey	The musical key the content of the track is in.
	LengthInMilliseconds	The length of the track (in milliseconds).
	Album	The album the track is on.
	OriginalAlbum	Name of album the track first appeared on.

1	OriginalFilename	First filename of the track (same as the Filename if it has not been renamed since created).
2	OriginalReleaseYear	The year the track was original released in.
3	FileOwner	Name of person with rights to this file.
4	Size	The size (e.g., in frames) of the track on the media.
5	ISRC	The International Standard Recording Code.
6	Software	Name of the software that ripped the track.
7	Rating	A rating for the track.
8	Comment	Comments regarding the track.
9	CDID	A hashed value of the disc identifier, allowing for faster lookups.
10	TOC	The disc identifier (based on the table of contents information) for the media storing the track.
11	Genre	The genre or category of the track.
12	Ripped	Whether the track has been ripped to a file.
13	OriginalIndex	The position on the album that the track appeared on.
14	LastEditedBy	An identification of the last entity (either the name of the meta data provider or "user") to edit any property associated with the object.
15	MediaTypeId	An identifier of the media type the track is on.
16	Attributes	A set of one or more flags that specify different play list attributes, such as a "deleted" indicating the play list is marked for deletion.
17	PlayCount	The number of times the track has been played.
18	Bitrate	The encoding rate of the file.
19	Protected	Flag indicating whether digital rights management (DRM) is used on the file.
20	CreationTime	The time the file was first added to the database.
21	LastModifiedTime	The last time any attribute of the track was changed in the database.
22	Composer	The composer of the track.
23	CoverArtSmall	A small graphic of the cover art of the media including the track.
24	CoverArtLarge	A larger graphic of the cover art of the media including the track.
25	TrackedFilename	An identifier of the file in the file system of the operating system using the database, allowing the file to be searched if it is not accessible by filename.
	Lyrics	The lyrics for the content of the track.

1 Batch Object

2 Property Name	3 Description
4 CDID	A hashed value of a disc identifier, allowing for faster lookups.
5 NumTracks	A number of tracks on the disc.
6 TOC	A disc identifier.

6 Link Object

7 Property Name	8 Description
9 Name	Friendly display name of a link.
10 URL	URL of the linked content.

10 Picture Object

11 Property Name	12 Description
13 Caption	A text description for a picture.
14 URL	A uniform resource locator of where the picture is located.
15 TrackedLink	An identifier of the picture in the file system of the operating system using the database, allowing the file to be searched if it is not accessible by filename.
16 Thumbnail	A thumbnail image for the picture.

17 CustomData Object

18 Property Name	19 Description
20 Name	The name of the object.
21 NumericValue	A numerical value for the object.
22 TextValue	A text value for the object.
23 DateValue	A date value for the object.
24 BinaryValue	A binary value for the object.
25 UnicodeTextValue	A unicode text value for the object.

1 ChildListEntry Object

2 Property Name	3 Description
4 ObjectType	5 GUID of the object type for the object.
5 ObjectInstanceID	6 Actual ID of the object.
6 ListPosition	7 Position of the media object in the play list.
7 ListInstanceID	8 Instance ID of the media object in the play list object.

9 DVD Object

10 Property Name	11 Description
11 Title	12 The title of the DVD.
12 Genre	13 The genre or category of the content on the DVD.
13 Director	14 The director of the content on the DVD.
14 ReleasedBy	15 The entity that released the DVD.
15 ProducedBy	16 The producer of the content on the DVD.
16 Studio	17 The studio that owns the content on the DVD.
17 Duration	18 The total playback time of the content on the DVD (or a portion of the content, such as a movie).
18 Rating	19 The rating of the content on the DVD.
19 RatingOrg	20 The organization that gave the rating for the content.
20 Copyright	21 Copyright information for the content on the DVD.
21 Comment	22 Comments regarding the DVD.
22 ReleaseDate	23 The release date of the DVD.
23 URL	24 Locator (URL) of the DVD for rendering.
24 ID	25 DVD identifier generated from a CRC 64 of portions of the DVD content.
25 LeadPerformers	26 The names of the lead performers in the content on the DVD.
26 BuyNow	27 Information on how to purchase the DVD.
27 MoreInfo	28 A URL to a web page with more information about the DVD.
28 CoverArtSmall	29 A small graphic of the cover art of the DVD.
29 CoverArtLarge	30 A larger graphic of the cover art of the DVD.
30 MPAA Rating	31 A rating assigned by the Motion Picture Association of America (MPAA) to the content of the DVD.

1 DVDTitle Object

2 Property Name	3 Description
4 Name	5 The name of the title (the primary content) of a DVD.
6 Genre	7 The genre or category of the title.
8 Director	9 The director of the title.
9 ReleasedBy	10 The entity that released the title.
10 ProducedBy	11 The producer of the title.
11 Studio	12 The studio that owns the title.
12 Duration	13 The total playback time of the title.
13 Rating	14 The rating of the title.
14 RatingOrg	15 The organization that gave the title the rating.
15 Copyright	16 Copyright information for the title.
16 Comment	17 Comments regarding the title.
17 ReleaseDate	18 The release date of the title.
18 LeadPerformers	19 The names of the lead performers in the title.
19 MPAARating	20 A rating assigned by the Motion Picture Association of America (MPAA) to the title.
20 ID	21 DVD identifier generated from a CRC 64 of portions of the DVD content.

22 DVDChapter Object

23 Property Name	24 Description
24 Name	25 The name of a chapter on a DVD.
25 Genre	26 The genre or category of the chapter.
26 Director	27 The director of the chapter.
27 ReleasedBy	28 The entity that released the chapter.
28 ProducedBy	29 The producer of the chapter.
29 Studio	30 The studio that owns the chapter.
30 Duration	31 The total playback time of the chapter.
31 Rating	32 The rating of the chapter.
32 RatingOrg	33 The organization that gave the chapter the rating.
33 Copyright	34 Copyright information for the chapter.
34 Comment	35 Comments regarding the chapter.
35 ReleaseDate	36 The release date of the chapter.
36 LeadPerformers	37 The names of the lead performers in the chapter.

1	MPAA Rating	A rating assigned by the Motion Picture
2	ID	DVD identifier generated from a CRC 64 of portions of the DVD content.

3

4 **Factoid Object**

5	Property Name	Description
6	Fact	The content (e.g., fact or comment) of the factoid.

7

8 Fig. 5 is a flowchart illustrating an exemplary process for rendering media

9 content meta data. The process of Fig. 5 is carried out by meta data manager 124

10 of Fig. 1, and may be implemented in software. The process of Fig. 5 is discussed

11 with reference to a CD, although other types of media may alternatively be used.

12 Initially, a disc is inserted to the computer (act 240). The disc is then

13 scanned for its table of contents information (act 242) and a disc identifier

14 generated based on the table of contents information (act 244).

15 A database lookup is then performed based on the disc identifier (act 246),

16 to determine whether the disc identifier exists in the database. In the example of

17 Fig. 3, this can be accomplished by checking the value for each entry in properties

18 table 206 that has a property ID for a disc identifier. If any of the values are the

19 same as the generated disc identifier, then the disc identifier already exists in the

20 database; otherwise, the disc identifier does not already exist in the database. The

21 result of this lookup is then used to determine the appropriate course of action (act

22 248).

23 If the disc identifier is in the database, then the meta data corresponding to

24 the disc identifier is obtained from the database (act 250) and rendered, or

25 otherwise made available, to the user (act 252). Returning to act 248, if the disc

1 identifier is not in the database, then a database representation is created for the
2 disc (act 254). This database representation includes the data that will be placed in
3 the tree structure (e.g., the disc identifier and an indication of each track on the
4 disc). The database representation is then converted into tree form (act 256), with
5 the disc identifier as the root node and a child node for each track on the disc. The
6 unknown meta data for each node object is initially identified as "unknown" (some
7 meta data, such as disc identifier, is known and can be stored in the database). A
8 remote server is then queried for the disc meta data (act 258). In the illustrated
9 example, this querying is performed by communicating the disc identifier to a
10 remote server(s), which in turn accesses the meta data it stores for any meta data
11 corresponding to the disc identifier, and returns that meta data to the requesting
12 computer. Processing then continues with walking the tree in order, updating any
13 changed meta data (act 260) to replace the values of "unknown" with the values
14 received from the server, and rendering the meta data (act 252).

15 It should be noted that the remote server may not always be accessible. For
16 example, the client 102 of Fig. 1 may not always be coupled to network 108 (e.g.,
17 the Internet), and thus not be able to access servers 104 and 106. In these
18 situations, meta data manager 124 keeps the values of "unknown" for the unknown
19 meta data and then accesses server 104 or 106 when the client is coupled to
20 network 108. At this point, the meta data is obtained from servers 104 and/or 106,
21 and the local meta data store updated appropriately. In one implementation, meta
22 data manager 124 maintains a separate record (e.g., in a separate file) of disc
23 identifiers for CDs that were added to the database when servers 104 and 106 were
24 not accessible to client 102. Thus, meta data manager 124 can readily ascertain
25 which meta data it needs from servers 104 and 106. Alternatively, no such

1 separate record may be maintained, and the database may be searched to identify
2 "unknown" values and determine, based on the tree structure, which disc
3 identifiers correspond to the needed meta data.

4 Fig. 6 is a flowchart illustrating an exemplary process for updating media
5 content meta data. The process of Fig. 6 is carried out by meta data manager 124
6 of Fig. 1, and may be implemented in software.

7 Initially, a user request to change the meta data is received (act 280), which
8 includes an indication of the change the user would like to make (for example,
9 what new data the user would like to add). The root object corresponding to the
10 meta data change is then identified (act 282), and selected as the initial object for
11 analysis (act 284). A determination is then made as to whether the selected object
12 is affected by the request (act 286). An object is affected by the request if the object
13 has meta data associated with it that corresponds to the user's request. For
14 example, the root disc object may have been affected if the user requests to change
15 the artist name of a CD, but not if the user requests to change the title of a track on
16 the CD.

17 If the selected object is affected by the request, then the meta data
18 associated with the selected object is updated based on the request (act 288). In
19 one implementation, this updating comprises overwriting previous meta data with
20 the meta data received as part of the user request. A check is then made as to
21 whether any additional objects are to be analyzed (act 290). In one
22 implementation, file manager 124 analyzes every object in the tree to determine
23 whether it is affected by the request. If there are one or more additional objects to
24 analyze, then one of the additional objects is selected (act 292) and the process
25

1 returns to act 286. However, if there are no more objects to analyze, then the
2 updating process ends (act 294).

3 The following example illustrates the process of updating media content
4 data of Fig. 6. Assume that a user changes the title for a song (track) on a CD that
5 is represented by track object 154 of Fig. 2. Meta data manager 124 begins with
6 disc object 150 and does not change any meta data associated with disc object 150.
7 However, track object 154 is affected by the user's requested change, so the title
8 meta data associated with track object 154 is changed to identify the new title.
9 Similarly, each of local file objects 160 and 162 are affected by the user's
10 requested change, so the title meta data associated with each of file objects 160
11 and 162 are changed to identify the new title.

12 Fig. 7 is a flowchart illustrating an exemplary process for generating meta
13 data for new media. The process of Fig. 7 is carried out by meta data manager 124
14 of Fig. 1, and may be implemented in software. The process of Fig. 7 is discussed
15 with reference to a CD, although other types of media may alternatively be used.

16 Many commonly-available computer systems allow a user to generate his
17 or her own media having whatever content he or she desires. For example, a
18 device commonly referred to as a CDROM burner allows a user to create his or
19 her own CDs with audio tracks of his or her choosing. This created CD can then
20 be played in virtually any conventional CD drive. However, because the CD is a
21 user-created CD, even though it may be created based on tracks from other
22 publicly available CDs, its table of contents and thus its disc identifier will be
23 different, and remote servers will not have meta data for the CD.

24 Meta data manager 124 solves this problem by communicating with the
25 module that is managing the creation of the new CD (e.g., content player 122).

1 The module managing creation of the new CD knows the identity of the tracks
2 being stored on the new CD. Meta data manager 124 uses this track identification
3 information to access the local meta data store and create new meta data. This
4 process is illustrated in Fig. 7.

5 Initially, a notification of a newly created disc is received (act 300). The
6 module managing creation of the new disc notifies meta data manager 124 that the
7 new disc is being created. Once the new disc is created, meta data manager 124
8 accesses the table of contents information for the new disc (act 302). A disc
9 identifier is then generated based on the table of contents information (act 304),
10 analogous to act 244 of Fig. 5.

11 A new root node is then created for the new disc in the local meta data store
12 (act 306). New children nodes are also created (act 308) based on the contents of
13 the disc. For each child (track) node, the manager 124 accesses the meta data for
14 the track stored in the database form the source of that track (act 310). In this
15 situation, the source of the track refers to the track (or file) that the user identifies
16 as being a track that should be stored on the new CD. This meta data is then
17 stored for the new disc (act 312).

18 It should be noted that, when creating a new CD, the tracks stored on the
19 new CD may optionally be associated with the source. In one implementation,
20 this is accomplished by identifying each track on the new CD as a child of the
21 original file or track. For example, if a user creates a new CD by copying a track
22 from a previous CD *X*, then a new CD root node and track node for the new track
23 are created as discussed above. Furthermore, the track for the new CD is
24 identified in the database as a child of the track on CD *X*, so any subsequent
25

1 changes to the meta data for that track on CD X are propagated to the meta data for
2 the new track.

3 Fig. 8 illustrates a more general exemplary computer environment 400,
4 which can be used to implement the improved meta data management described
5 herein. The computer environment 400 is only one example of a computing
6 environment and is not intended to suggest any limitation as to the scope of use or
7 functionality of the computer and network architectures. Neither should the
8 computer environment 400 be interpreted as having any dependency or
9 requirement relating to any one or combination of components illustrated in the
10 exemplary computer environment 400.

11 Computer environment 400 includes a general-purpose computing device in
12 the form of a computer 402. Computer 402 can be, for example, any of computing
13 devices 102, 104, or 106 of Fig. 1. The components of computer 402 can include,
14 but are not limited to, one or more processors or processing units 404, a system
15 memory 406, and a system bus 408 that couples various system components
16 including the processor 404 to the system memory 406.

17 The system bus 408 represents one or more of any of several types of bus
18 structures, including a memory bus or memory controller, a peripheral bus, an
19 accelerated graphics port, and a processor or local bus using any of a variety of
20 bus architectures. By way of example, such architectures can include an Industry
21 Standard Architecture (ISA) bus, a Micro Channel Architecture (MCA) bus, an
22 Enhanced ISA (EISA) bus, a Video Electronics Standards Association (VESA)
23 local bus, and a Peripheral Component Interconnects (PCI) bus also known as a
24 Mezzanine bus.

1 Computer 402 typically includes a variety of computer readable media.
2 Such media can be any available media that is accessible by computer 402 and
3 includes both volatile and non-volatile media, removable and non-removable
4 media.

5 The system memory 406 includes computer readable media in the form of
6 volatile memory, such as random access memory (RAM) 410, and/or non-volatile
7 memory, such as read only memory (ROM) 412. A basic input/output system
8 (BIOS) 414, containing the basic routines that help to transfer information
9 between elements within computer 402, such as during start-up, is stored in ROM
10 412. RAM 410 typically contains data and/or program modules that are
11 immediately accessible to and/or presently operated on by the processing unit 404.

12 Computer 402 may also include other removable/non-removable,
13 volatile/non-volatile computer storage media. By way of example, Fig. 8
14 illustrates a hard disk drive 416 for reading from and writing to a non-removable,
15 non-volatile magnetic media (not shown), a magnetic disk drive 418 for reading
16 from and writing to a removable, non-volatile magnetic disk 420 (e.g., a "floppy
17 disk"), and an optical disc drive 422 for reading from and/or writing to a
18 removable, non-volatile optical disc 424 such as a CD-ROM, DVD-ROM, or other
19 optical media. The hard disk drive 416, magnetic disk drive 418, and optical disc
20 drive 422 are each connected to the system bus 408 by one or more data media
21 interfaces 426. Alternatively, the hard disk drive 416, magnetic disk drive 418,
22 and optical disc drive 422 can be connected to the system bus 408 by one or more
23 interfaces (not shown).

24 The various drives and their associated computer-readable media provide
25 non-volatile storage of computer readable instructions, data structures, program

1 modules, and other data for computer 402. Although the example illustrates a
2 hard disk 416, a removable magnetic disk 420, and a removable optical disc 424, it
3 is to be appreciated that other types of computer readable media which can store
4 data that is accessible by a computer, such as magnetic cassettes or other magnetic
5 storage devices, flash memory cards, CD-ROM, digital versatile discs (DVD) or
6 other optical storage, random access memories (RAM), read only memories
7 (ROM), electrically erasable programmable read-only memory (EEPROM), and
8 the like, can also be utilized to implement the exemplary computing system and
9 environment.

10 Any number of program modules can be stored on the hard disk 416,
11 magnetic disk 420, optical disc 424, ROM 412, and/or RAM 410, including by
12 way of example, an operating system 426, one or more application programs 428,
13 other program modules 430, and program data 432. Each of such operating
14 system 426, one or more application programs 428, other program modules 430,
15 and program data 432 (or some combination thereof) may implement all or part of
16 the resident components that support the distributed file system.

17 A user can enter commands and information into computer 402 via input
18 devices such as a keyboard 434 and a pointing device 436 (e.g., a "mouse").
19 Other input devices 438 (not shown specifically) may include a microphone,
20 joystick, game pad, satellite dish, serial port, scanner, and/or the like. These and
21 other input devices are connected to the processing unit 404 via input/output
22 interfaces 440 that are coupled to the system bus 408, but may be connected by
23 other interface and bus structures, such as a parallel port, game port, or a universal
24 serial bus (USB).

1 A monitor 442 or other type of display device can also be connected to the
2 system bus 408 via an interface, such as a video adapter 444. In addition to the
3 monitor 442, other output peripheral devices can include components such as
4 speakers (not shown) and a printer 446 which can be connected to computer 402
5 via the input/output interfaces 440.

6 Computer 402 can operate in a networked environment using logical
7 connections to one or more remote computers, such as a remote computing device
8 448. By way of example, the remote computing device 448 can be a personal
9 computer, portable computer, a server, a router, a network computer, a peer device
10 or other common network node, and the like. The remote computing device 448 is
11 illustrated as a portable computer that can include many or all of the elements and
12 features described herein relative to computer 402.

13 Logical connections between computer 402 and the remote computer 448
14 are depicted as a local area network (LAN) 450 and a general wide area network
15 (WAN) 452. Such networking environments are commonplace in offices,
16 enterprise-wide computer networks, intranets, and the Internet.

17 When implemented in a LAN networking environment, the computer 402 is
18 connected to a local network 450 via a network interface or adapter 454. When
19 implemented in a WAN networking environment, the computer 402 typically
20 includes a modem 456 or other means for establishing communications over the
21 wide network 452. The modem 456, which can be internal or external to computer
22 402, can be connected to the system bus 408 via the input/output interfaces 440 or
23 other appropriate mechanisms. It is to be appreciated that the illustrated network
24 connections are exemplary and that other means of establishing communication
25 link(s) between the computers 402 and 448 can be employed.

1 In a networked environment, such as that illustrated with computing
2 environment 400, program modules depicted relative to the computer 402, or
3 portions thereof, may be stored in a remote memory storage device. By way of
4 example, remote application programs 458 reside on a memory device of remote
5 computer 448. For purposes of illustration, application programs and other
6 executable program components such as the operating system are illustrated herein
7 as discrete blocks, although it is recognized that such programs and components
8 reside at various times in different storage components of the computing device
9 402, and are executed by the data processor(s) of the computer.

10 Computer 402 typically includes at least some form of computer readable
11 media. Computer readable media can be any available media that can be accessed
12 by computer 402. By way of example, and not limitation, computer readable
13 media may comprise computer storage media and communication media.
14 Computer storage media includes volatile and nonvolatile, removable and non-
15 removable media implemented in any method or technology for storage of
16 information such as computer readable instructions, data structures, program
17 modules or other data. Computer storage media includes, but is not limited to,
18 RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM,
19 digital versatile discs (DVD) or other optical storage, magnetic cassettes, magnetic
20 tape, magnetic disk storage or other magnetic storage devices, or any other media
21 which can be used to store the desired information and which can be accessed by
22 computer 402. Communication media typically embodies computer readable
23 instructions, data structures, program modules or other data in a modulated data
24 signal such as a carrier wave or other transport mechanism and includes any
25 information delivery media. The term "modulated data signal" means a signal that

1 has one or more of its characteristics set or changed in such a manner as to encode
2 information in the signal. By way of example, and not limitation, communication
3 media includes wired media such as wired network or direct-wired connection,
4 and wireless media such as acoustic, RF, infrared and other wireless media.
5 Combinations of any of the above should also be included within the scope of
6 computer readable media.

7 The invention has been described herein in part in the general context of
8 computer-executable instructions, such as program modules, executed by one or
9 more computers or other devices. Generally, program modules include routines,
10 programs, objects, components, data structures, etc. that perform particular tasks
11 or implement particular abstract data types. Typically the functionality of the
12 program modules may be combined or distributed as desired in various
13 embodiments.

14 For purposes of illustration, programs and other executable program
15 components such as the operating system are illustrated herein as discrete blocks,
16 although it is recognized that such programs and components reside at various
17 times in different storage components of the computer, and are executed by the
18 data processor(s) of the computer.

19 Alternatively, the invention may be implemented in hardware or a
20 combination of hardware, software, and/or firmware. For example, one or more
21 application specific integrated circuits (ASICs) could be designed or programmed
22 to carry out the invention.

1 **Conclusion**

2 Although the description above uses language that is specific to structural
3 features and/or methodological acts, it is to be understood that the invention
4 defined in the appended claims is not limited to the specific features or acts
5 described. Rather, the specific features and acts are disclosed as exemplary forms
6 of implementing the invention.

7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25